Configure Mozilla Firefox for DoH

If you use Mozilla Firefox as your browser, you can configure DNS-over-HTTPS by:

Preferences menu:

1. In menu, navigate to **Tools** > **Preferences** > **General** > **Settings** > **Network Settings.**
2. Select **Enable DNS over HTTPS**.
3. Choose any of the following to set DNS resolver:
    - (Default) Set Cloudfare.
    - Set a custom DNS resolver.

Configuration panel:

1. Type **about:config** in the URL bar.
2. In the search area, type **network.ttr.mode** and set any of the following values:
    - **2:** Turns on DoH and keeps the regular DNS only as a backup.
    - **3:** Turns on DoH with no regular DNS support.
3. In the search area, type **network.ttr.uri** to set the domain name of the DNS resolver.
    - The default Cloudfare URI is *https://mozilla.cloudflare-dns.com/dns-query*
4. (Optional). In the search area, type **network.ttr.bootstrapAddress** to set up IP address.
    1. Type **1.1.1.1** for **Cloudfare**.
    2. Type **8.8.8.8** for **Google**.

Security: Surf web via a secure DNS server

There are a number of ways to safely browse the internet. One could use a proxy server, TOR proxy, VPN, or use an operating system that uses those principles. TAILS and Qubes are the popular ones that comes to my mind. All these solutions are available and things are sorted if you are a techie. For an average user, security has to bundled by default. Until 2005, the internet engineering task force (IETF) was not at all interested in securing the very basic such as DNS lookup.

A DNS lookup (domain name to IP address mapping) is a plain text query over a UDP or TCP connection on port 53. DNS cache poisoning is one of many attacks possible in this scenario. In 2005, request for comments 4033 (RFC4033) was put forward for consideration. The Domain Name System Security Extensions (DNSSEC) was introduced. The intention was to authenticate the origin of data and maintain its integrity. It would mean to create a chain of authentication and each DNSSEC resolver has to sign its resource records with its public key and records have to be validated upto root resolver. This could fall into a bogus (invalid trust chain), indeterminate (some portion is valid), insecure (trust chain breaks at a non-existent DS record), and secure (a valid full trust chain) category. The major drawback is that all ISPs/registrars have to agree to support it. Plus, there’s always a threat of denial of service attacks. An attacker can use DNSSEC mechanism itself to send updated messages and force security-aware name server to re-sign the resource record sets (RRsets). The request and response queries are not secure at all. Some proposals exist such as DNSCurve, DNSCRYPT, Confidential DNS, and IPSECA. These measures if utilized along with DNSSEC could provide some level of security. Additionally, a proposal came for using DNS over Datagram Transport Layer Security (DTLS) DNSoD. There have been several updates to that RFC itself since then but none of them provide a comprehensive secure solution.

A compiled version of security concerns is available for reference, RFC7626. It summarizes what all needs to be taken care of. Following that, in 2016, DNS over Transport Layer security (DoT) was proposed (RFC7858). In this, requests and responses are encrypted with a TLS handshake between a stub (client) and recursive-name resolver (server) over port 853 by default. The initial connection is over TCP and after successful connection TLS handshake is required. The session could be over any other port provided client and server are configured as such. However, there’s a catch, a stub (client) doesn’t necessarily need to authenticate the server. This opens the door to man-in-the-middle (MitM) attacks.

In 2018, DNS over HTTPS (DoH) was put forward for comments in RFC8484. This introduces the same principles of DNS-over-TLS but also mandates that the server needs to be authenticated. The connection is established over HTTPS on a regular HTTPS port 443. By far, DoH is the most secure way to browse the internet. The proposal was partly put forward by Mozilla and the very reason it’s my favorite internet browser. Mozilla has provided its own wiki (Trusted Recursive Resolver) to guide users how to setup DoH.